Tema	Horas	Descripción
Sintáxis Básica de JavaScript: - Tipos - Condicionales, bucles - Objetos - Tipos vs Objetos	2	Empezamos con la parte más básica, mostrando algunas particularidades propias del lenguaje.
Funciones: - Constructores - Prototype	2	La estructura más importante del lenguaje, permiten programar orientado a objetos, aplicar herencia y extender objetos.
Patrones: - Closures - Módulos - Mixins - Publish / Subscribe	3	Un primer vistazo a técnicas que permiten crear código fácilmente mantenible.
DOM - CSSOM: - Manipulación de nodos - Eventos - Box Model - Screen - Media Queries	3	Un rápido paseo por el DOM y el nuevo CSSOM. Manipular elementos, agregar eventos, entender el box model y ver el lado JavaScript de las hojas de estilo.
APIs del navegador: - File - History - Storage - Websocket	2	Entenderemos algunas de las APIs de HTML5 que permiten pasar de sitios web a aplicaciones web: Leer archivos antes de subirlos, manejar el historial del navegador, verificar la orientación del dispositivo, guardar datos en local y recibir mensajes del servidor automáticamente.
Peticiones asíncronas: - Objeto XmlHttpRequest - JSON - Subir archivos de manera asíncrona - Promises	3	La forma nativa de leer y enviar datos al servidor sin refrescar el navegador. También veremos la forma de mejorar la forma de manejar estas peticiones con promises.
Pruebas: - Jasmine - Mocha	2	Uno de los pilares del desarrollo en general, ahora con JavaScript. Pruebas automatizadas de funciones y pequeños módulos.
Taller: Creación de biblioteca JavaScript	2	Aplicaremos todo lo aprendido en la primera parte del módulo creando una biblioteca para manipular el DOM y una biblioteca para leer información de servicios externos.
Aplicaciones medianas con JavaScript: - Módulos con RequireJS - Namespacing - Uso de bibliotecas de terceros	2	Pasamos de crear scripts a crear aplicaciones organizando el código en forma de módulos.
Aplicaciones MVC/MV*: - Patrón MVC - Arquitectura REST - Implementación con JavaScript nativo	3	Siguiente paso en la creación de aplicaciones mantenibles. El patrón MVC separa responsabilidades entre módulos de código, mientras que la arquitectura REST define operaciones básicas con recursos externos.
jQuery: - Selectores - Eventos - $.ajax	2	El estándar de facto para manipular el DOM, eventos y realizar peticiones asíncronas.
Backbone.js I: - Modelos y Colecciones - Eventos - Vistas - Routers	2	Una de las primeras bibliotecas que utilizó el patrón MVC, en su variante MV*. Veremos los conceptos propios de Backbone y como se comunican entre sí.

Backbone.js II: - Performance - Backbone.Marionette	2	Mejoramiento en el código para obtener una aplicación más rápida. Manejo de componentes más complejos con Marionette.
Taller: Creación de SPA con Backbone.js	2	Utilizamos jQuery y Backbone.js para crear una single-page application con comunicación a un servidor externo vía REST y JSON.